

OS-9 Newsletter

Volume II No. 12 <<< BELLINGHAM OS-9 USERS FORUM >>> December 31, 1991

Basic09 Part-3

Tutorial by Scott Honaker & Rodger Alexander

Last month you were given the Basic09 source code for the MAIN MENU, VIEW RECORD, CREATE DATABASE, and ADD RECORD Procedures. I also gave you the parameters needed to write the OPEN DATABASE Procedure so that you might try your hand at writing your own procedure.

I received at least one phone call (Skip McNealy, Concrete, WA), so I know that at least a few of you are getting into this.

This month I'll include a print-out of my OPEN DATABASE Procedure and give you the parameters and a couple of hints about designing the remaining Procedures.

REVIEW:

The MAIN MENU Procedure does pretty much what the title says (Provides all of the options in a menu format).

From the MAIN MENU you can <C>reate a database file and <A>DD records to your database file. However, when you exit Basic09 and then later come back to <V>iew or <A>dd more records to your database.... You won't be able to do it because you have no OPEN DATABASE Procedure.

PROCEDURE OpenDB

```
TYPE address= FName:STRING[10]; LName:STRING[15]; address1:STRING[20];  
address2:STRING[20]; city:STRING[15]; state:STRING[2]; zip:STRING[10];  
phone:STRING[14]
```

```
PARAM DBPath:BYTE; Top:INTEGER; Current:INTEGER; DBName:STRING[32]
```

```
DIM rec:address
```

```
DIM option:STRING[1]
```

```
IF Top>0 THEN CLOSE #DBPath \ ENDIF
```

```
Top=0
```

```
10 PRINT CHR$(12) \ INPUT "Enter filename: ",DBName
```

```
ON ERROR GOTO 100
```

```
OPEN #DBPath,DBName
```

```
WHILE NOT(EOF(#DBPath)) DO
```

```
SEEK #DBPath,SIZE(rec)*Top
```

```
Top=Top+1
```

```
ENDWHILE
```

```
Top=Top-1
```

- IN THIS ISSUE -

Basic09 Database (Part-3)	1
Rainbow Ads CHEAP!	2
Ten Commandments for C	3
PWD/PXD--Mysteries revealed	3
Bernie's Bit Bucket "What The Rainbow didn't tell you"	5
Screen Saver script file	7
G- Windows for 68K OS9	8
Club Activities Report	11

MAIN MENU

Current Database:

Record number: of Records

```
Main Menu:  
<C>reate database <D>open database Pac<K> database  
<A>dd record <D>elete record <E>dit record  
<P>rint record Find<N>ext record <S>ort database  
<P>rint database Print <L>abels <W>rite records  
Esc<X>it program
```

Menu Choice:

```
Current=Top
```

```
END
```

```
100 IF ERR=216 THEN
```

```
PRINT CHR$(12) \ PRINT \ PRINT
```

```
PRINT "File: "; DBName; " does not  
exists"
```

```
PRINT \ PRINT "Do you wish to try again
```

SUBSCRIPTION INFORMATION

The OS-9 Newsletter is compiled and printed monthly by the Bellingham OS-9 Users Group. Subscription rates are \$5 for 6 months or \$10 for 12 months. Mail your subscription check to:

Rodger Alexander
3404 Illinois Lane
Bellingham, WA. 98226

BENEFITS TO SUBSCRIPTION MEMBERS

As a subscribing member of the Bellingham OS9 Users Group, you enjoy the following benefits:

1. Monthly Newsletter
2. Public Domain Library (6-Megs)
3. Technical assistance (734-5806)
4. FREE Classified Ads
5. On-Line OS-9 Conference (676-5787)

```
(Y/N)? ";
200 RUN inkey(option)
IF option="" THEN 200
ELSE
IF option="Y" OR option="y" THEN
GOTO 10
ELSE CLOSE #DBPath
ENDIF
ENDIF
END
```

NOTE: Remember to save your basic09 source code files using the "" options:

SAVE* DataBase

This will merge all of the source files you have in the Basic09 buffer to a single file on your disk.

Later, when you "LOAD" your DataBase file, the individual Procedures will list out separately.

Now run the database program from Basic09 and "<O>pen" a database file that you previously created. Use the arrow keys to "page" through your records.

DELETE RECORD:

This procedure will allow the current record to be deleted by setting the Name field equal to "" (null string). This will leave the database with an empty record in the middle which will later have to be packed and removed. In theory, this should be the easiest of all of the procedures. However, you might consider the screen appearance. I wanted to keep the Main Menu Options in tact while each record is viewed in the top left corner of the screen, so I wrote the DEL RECORD Procedure very similar to the VIEW Procedure with the exception of a prompt to ask if the current record should be deleted or not, and a couple of lines to re-assign "" (null string) to the "rec.FName" and "rec.LName" fields. Then you need to write or "PUT" the new values into the database in order to replace the original data:

```
INPUT "Delete this record: (Y/N)? ",option
IF option="Y" OR option="y" THEN
    rec.FName=""
    rec.LName=""
    PUT #DBPath,rec.FName
    PUT #DBPath,rec.LName
ENDIF
```

OOOPS! ALMOST FORGOT: One thing left to do to make this delete option work correctly. You will have to modify the VIEW RECORD Procedure so that it will be sensitive to the delete function and respond properly. Simply add one line of code about 7 lines up from the bottom.

```
IF rec.FName="" THEN 10
Insert this line just before the "PRINT rec.address1"
```

line. This will allow the view option to skip over the record you deleted. But, keep in mind that the record is still there and eventually will have to be removed or PACKed.

NEXT MONTH: In the January's installment we will tackle one of the most difficult Procedures: EDIT RECORD. We're going to need some help for this one...OK JEFF?

"CHEAP" Ads in the Rainbow!?

Small software vendors don't advertise in Rainbow much anymore. Why? - the rates are just too much for a small company, and the market is not getting larger. The problem is, the only real advertising outlet for Color Computer software is the Rainbow- with the high advertising rates.

With this in mind, I have made arrangements with Rainbow to provide a classified ad page. I will be totally responsible for payment to Rainbow, and in providing the ad copy. Page will be in three columns of text only ads with a minimum of 6 lines and 38 characters across each line.

Ad copy may be changed from month to month. I will retain the right to edit ads to fit. Any mistakes will be run over at no charge for the number of months the mistake ran. Telephone changes will be allowed, BUT FARNA WILL NOT BE RESPONSIBLE FOR ANY MISTAKES on telephone updates. I will also accept changes via Delphi E-Mail, but they MUST be sent well ahead of the deadline. Ads are typeset locally then forwarded to Rainbow camera ready.

Personal checks or money orders will be accepted. ADS MUST BE PREPAID FOR A MINIMUM OF THREE MONTHS PAID IN ADVANCE (\$89.10 minimum charge).

I hope to start implementing this new advertising idea in the April 1992 Rainbow. I will need ad copy (preferably typed, dot matrix draft printout is acceptable) and payment by 19 January for the ads to start appearing in the April issue. If enough ads to fill 1/2 page are not received due to this short notice, ads will start appearing in the May issue. No checks/money orders will be cashed until the minimum number of ads are received. Response has been good so far. I expect to have at least one full page for May.

As you know, just a simple ad inviting inquiries for a catalog, and maybe listing just your newest offering(s), will be noticed after a few months in Rainbow. That little bit of exposure could improve your sales tremendously. For more information, contact:

FARNA Systems
Francis G. Swygert
904 2nd Avenue
Warner Robins, GA 31098
Phone 912-328-7859
Delphi E-Mail - DSRTFOX

Ten Commandments for C Programers

by Henry Spencer

Thou shalt run lint frequently and study its pronouncements with care, for verily its perception and judgement oft exceed thine.

Thou shalt not follow the NULL pointer, for chaos and madness await thee at its end.

Thou shalt cast all function arguments to the expected type if they are not of that type already, even when thou art convinced that this is unnecessary, lest they take cruel vengeance upon thee when thou least expect it.

If thy header files fail to declare the return types of thy library functions, thou shalt declare them thyself with the most meticulous care, lest grievous harm befall thy program.

Thou shalt check the array bounds of all strings (indeed, all arrays), for surely where thou tighest "foo" someone someday shall type "supercalifragilisticexpialidocious".

If a function be advertised to return an error code in the event of difficulties, thou shalt check for that code, yea, even though the checks triple the size of thy code and produce aches in thy typing fingers, for if thou thinkest "it cannot happen to me", the gods shall surely punish thee for thy arrogance.

Thou shalt study thy libraries and strive not to re-invent them without cause, that thy code may be short and readable and thy days pleasant and productive.

Thou shalt make thy program's purpose and structure clear to thy fellow man by using the One True Brace Style, even if thou likest it not, for thy creativity is better used in solv-

ing problems than in creating beautiful new impediments to understanding.

Thy external identifiers shall be unique in the first six characters, though this harsh discipline be irksome and the years of its necessity stretch before thee seemingly without end, lest thou tear thy hair out and go mad on that fateful day when thou desirest to make thy program run on an old system.

Thou shalt foreswear, renounce, and abjure the vile heresy which claimeth that "All the world's a VAX", and have no commerce with the benighted heathens who cling to this barbarous belief, that the days of thy program may be long even though the days of thy current machine be short.

PWD/PXD Mysteries

Copyright (c) 1991 Zack C. Sessions

OK, so there isn't an OS-9 System Call which returns your current data or execution directory. So, just how do pwd and pxd do it? This document will attempt to explain that so that you will understand it. Also, understanding how pwd and pxd work will also further your knowledge of OS-9 Disk Structure.

Let us begin our discussion with the pwd command, first. Once it's operation is explained and understood, the operation of pxd will be a relatively simple matter to understand.

First off, a small discussion on OS-9 Disk Structure. The basic entity which is contained on any disk, OS-9 or otherwise is known as a file. Actually, the information stored on a disk is stored in fixed length segments normally referred to as sectors. The operating system involved

organizes information stored in these sectors into discernable units. These are the files contained on that disk. Special information is stored by the operating system on the disk which indicate to the operating system how many files there are on a disk, what their names are, where they are located on the disk, and how large they are. This is the basic amount of information which would be required.

OS-9 and other operating systems also store various other information about each file such as the date of creation, date last modified, and so forth. With Disk Extended Color Basic, this information was stored in part of Track 17. In this manner, you are limited to the number of files the disk can store by the amount of space allocated to this special storage area.

OS-9 and other operating systems allows for a greater capacity by storing this "overhead" file information in areas which are dynamically allocated. These areas are normally referred to as "directories" and their related data structures.

A directory is nothing more than a special file manipulated only by the operating system. In the case of OS-9, there is also a special sector allocated to each file to store special information. This sector is called the file descriptor sector. In actuality, with OS-9, the only information stored in the directory file is the name of the files it contains and for each file a pointer to the file's file descriptor sector. File descriptor sectors exist on a one to one correspondence with all files on the disk.

A file's file descriptor segment can be anywhere on the disk, it's actual location is irrelevant in regards to the relative position of the directory which contains the file's name or to the data which the file contains. One special characteristic of the OS-9 disk structure which is shared by several other operating systems is that a directory can contain a directory. This is possible since a directory is nothing more than a file. Therefore a direc-

tory file, just as any other file has its very own unique file descriptor sector. This means that the directory structure on a disk can take on a hierarchical structure, a structure which can be compared to a tree. The leaves are the individual non-directory files. The branches are the directories. All this has to start somewhere and as in a tree this special directory is called the "root" directory. It is the ultimate parent of all other directories on the disk. The directory file which contains another directory is referred to as the "parent" directory of that directory.

The name of the root directory is the same as the name of the device on which it resides. Thus the name of the root directory on a hard disk who's name is /h0 is /h0. This concept is a little more confusing with a floppy disk drive. If you have two floppy disk drives, /d0 and /d1, then the name of the root directory of a floppy disk which is in device /d0 is /d0, but the name of the root directory of that very same floppy disk when the disk is placed in device /d1 is /d1. Root directories are a little more capable than that of a tree, since it can contain not only directories (branches) but also non-directory files (leaves).

The root directory is automatically created on a disk when the disk is formatted. subsequent directories which are created on the disk are created by the OS-9 System call I\$MakDir. A simple user interface to this system call is provided with the Shell command makdir. The first directories created on a disk must first reside in the root directory. Once a directory is created in the root directory, a directory can then be created in that directory. Any or all of these directories can contain non-directory files or other directories.

A lot of this may be review for many readers of this file, but please bear with me I am leading up to the important point here. That is, every directory always contains two special files which are created automatically

by the I\$MakDir System Call. These are "hidden" files since the Shell dir command doesn't report their existence, but there are there nonetheless. Actually, they aren't really files, but merely special entries in the directory file. This will become apparent shortly.

The names of these entries are "." and "..". The names of these entries do not actually contain the surrounding quotation marks, they names are just the periods. I will surround them with quotation marks to separate them for clarity. These entries could be better referred to as "pointers" to files which already exist. The entry "." is a pointer to the directory itself and the entry ".." is a pointer to the parent directory of the directory. Maybe an example will help to clear up any questions. Let's say we have a disk named /h0. It's root directory's file descriptor sector is located at LSN 25. (I'll keep this simple and use all decimal numbers for LSN, but in actuality LSNs are stored as 3 byte binary integers.)

Now, let's say we create a directory in the root directory called /h0/TEXT. Let's also say that the file descriptor sector for the directory is located at LSN 50. The /h0/TEXT directory will automatically have those two special entries "." and "..". The LSN of the file descriptor sector for the "." entry will be 50 and the LSN for the file descriptor sector for the ".." entry will be 25. An important thing to remember is that the entry for "." and ".." for the root directory are the SAME, because the parent of the root directory is the root directory itself.

These entries are really "synonym" file names. So, if your current data directory is /h0/TEXT and you ask for a directory of file "..", then you are really asking for a directory of /h0. The entries "." and ".." can be used ANYWHERE that a directory name can be used. For example, you can also "chd .." to change your data directory to the parent of the directory you are current in.

Now, just how does pwd make use

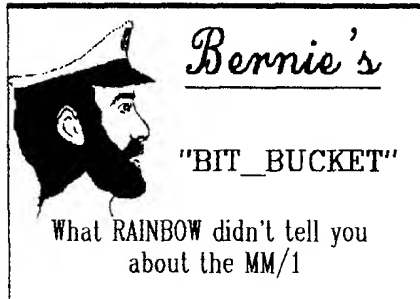
of this information to determine your current data directory? Well, pwd first opens the directory file "...", the current data directory, and reads the entries "." and "..". This gives pwd the LSN of the file descriptor sector for the parent directory and the current directory. If these are equal, then pwd has finished it's job, it is now at the root directory. If they are unequal, then we need to determine the name of the "." directory. This is the main processing loop for pwd. The name of the "." directory is done by changing the current data directory to the directory "...". It then checks to see if the LSNs for "." and ".." are equal. If they are, pwd is done. If they are unequal, it reads through the directory searching for an entry whose file descriptor sector LSN is equal to the file descriptor sector LSN for the previous "." entry. When that is found, we have the name for the previous "." entry. That name saved away. We now have the file descriptor sector LSN for this "." entry, so we start the process over by changing directory to "." again. When it finally finds a directory whose file descriptor sectors LSN are equal, it determines the name of the device with a I\$GetStt SS.DevNm System Call and displays the completed result.

You'll probably have to read the previous paragraph a few times before it becomes clear to you just how this process is done. Once you understand it, it will be obvious to you that to perform the similar pxd command, all that needs to be done is to access the directories with the execution bit set! In fact if you compare the pwd program with the pxd program, there are only 4 bytes which are different! (Not counting the 3 CRC bytes, of course!) And two of those 4 are the internal names of the programs! So actually, there are only two bytes which are effectively different for the two programs and these are the two bytes which control the access mask for the I\$Open of the "." directory and the I\$ChgDir to the ".." directory.

If you can read C, I have included

the equivalent code in C for the `pwd` command and the `pxd` command. While it would be extremely inefficient for you to compile and actually use these programs they are included for instructional purposes only. See, the original `pwd` and `pxd` commands are written in ASM, which makes their binaries much smaller! If you wish to incorporate the code from these two programs in a program you are writing to avoid forking `pwd` and/or `pxd` commands, you are most welcome to do so! They were written in a form as to NOT require any special library, they will compile and work just fine with the stock `clib.l` which comes with the Microware C Compiler for OS-9/6809. In fact, even though `pwd` could have been written slightly simpler using the Kreider Lib, `pxd` would not even be possible, since there is no option with the `opendir()` function to open the directory with the execution bit set, which is required by the `pxd` function.

I hope this file and the associated C program sources has helped increase your knowledge and understanding of OS-9 and its disk file structure.



This is a user report of what the advertisements for the MM/1 led me to expect, why I purchased an MM/1, and the history of my dealings with Interactive Media Systems (IMS), the company that makes the MM/1. Pre-Market Publicity:

Some well known names in the CoCo/OS-9 community began advertising in RAINBOW in 1989 (using the name KENNETH-LEIGH ENTERPRISES) asking for input on what they could build for those who wanted a CoCo-4. They even had a

"tear out" questionnaire with many features listed and asked how many of these features we wanted and how much we would pay for them. Very good market research technique. I wrote a letter (I refuse to tear up my RAINBOW magazines) expressing my own desires, one of which was expandability. I got a nice letter in return. In March 1990 K-L Enterprises advertised in RAINBOW that they would introduce the computer we demanded in 1990. Later, it was reported that the computer was already in the design stages when the questionnaire was printed so it seems that little input was gained other than names of potential customers and direction on how best to advertise the computer to appeal to customers' needs.

Somewhere along the way the company changed the name to Interactive Media Systems (IMS). In July 1990 MM/1 was claiming to be able to "put us in the mainstream" by being able to run "thousands of MS-DOS programs" on our OS-9/68K MM/1 and be able to expand to 9 Megabytes of RAM. In August 1990 the MM/1 was advertised for sale. Their ad in RAINBOW stated, for \$###.## you get an MM/1 with.... and then listed the different packages of things they were offering for sale. As late as December 1990 MM/1 was still saying they would be offering MS-DOS and UNIX application compatibility. They then began modifying their claim to compatibility (if you had believed their original ads you, like I did, would not have noticed the change) to having applications "brought over from MS-DOS and UNIX." They were still strongly advertising "pouring our energy into bringing the MM/1 the best of the MS-DOS and UNIX software catalogs" and also strongly implying a high degree of CoCo hardware compatibility.

Their January 1991 ad in RAINBOW said, "You can do it all on the MM/1." I submit that the word "you" in this instance means "anyone who purchases an MM/1" and that the word "can" is present tense. Used

together I believed these to mean that if I sent them some money they would send me a computer by return mail and I would be able to do all these wondrous things mentioned in their ad. Last year's KIT SALES: In October 1990 IMS began advertising that they would send you a 1-Meg RAM kit or a 3-Meg RAM kit. They said (on the BBS's) that they couldn't sell completed computers until they got FCC approval. This was supposed to help keep the company going and allow us to begin using the MM/1 at an earlier date than we would if they made us wait until the mean old FCC finally got around to looking at their nice new computer. I ordered a 3-Meg kit. Price - \$1149.00. The date was February 28, 1991.

I called them many times over the next eight months and was told that they couldn't send it right now because -

- a. I was too low on the list of people who had ordered their popular, new computer - first come, first served.
- b. They had a problem with a chip supplier and couldn't get the chips to use with the board that they had been given to develop the prototype.
- c. They had a small redesign problem because of a change in chips.
- d. My catalogs should be shipped by the end of July. What catalogs?
- e. "There are only three of us trying to run this company and the one who does the shipping had a family emergency and has been gone for a couple of weeks. I was under the impression that your computer was ready to ship, so it must be waiting for him to get back."
- f. "We are shipping the I/O boards now! I'm looking at Bob van der Poel's I/O board (2-Meg upgrade) right now. It is here on my desk!" (This was in AUGUST 1991 and Bob told me that his prototype MM/1 didn't even have a provision for adding an I/O board and he could think of no reason why I should be told that one was ready to ship to him.) Remember, the first ads for the sale of MM/1 computers were in

August of 1990.

I was given a delivery date for my brand new MM/1 three different times. The last time they carried thru and delivered a kit. In October 1991, 8 months after I sent them the money, 14 months after they first offered the machine for sale. Furthermore, they had not sent the I/O board with it's additional 2-megs of RAM and ability to attach my SCSI hard drive.

There were many things missing from the "kit". Here are a few - a complete list of what was in the kit, a warrantee card, OS-9/68K and MicroWare Basic documentation, the I/O board and 2 Megs of additional RAM, and clear directions of how to assemble the "kit". VERY notably missing from the papers that came with the kit were the pin-outs for the "high speed serial port" on the main buss, the /t0 serial port, and, most of all, the VIDEO port! An inspection shows that they didn't even bother to mark pin #1 of the headers! You get to guess which way to attach cables! This is worse than just sloppy workmanship. I've put together many kits in my life. All kits have directions on how to assemble them. I've also put together many puzzles. The difference between a "kit" and a "puzzle" is whether or not there are directions. In this case, since there are no directions, I paid for a kit but IMS delivered only part of a puzzle. Some puzzles I enjoy, but I don't enjoy paying \$1149.00 for a falsely advertised puzzle.

Since I had (fortunately) gotten the pin-out for the video port from Paul Ward several months before so I could have my MM/1 -to- 8CM515 cable ready I was able to hook up the monitor anyway, NO thanks to the "directions". Again, this is worse than just sloppy workmanship. When I finally got the MM/1 plugged in I got a blue screen with the OS-9 68000 BootStrap Module sign-on message and NOTHING ELSE. Then I quit getting even that. By this time I was so disgusted with the quality of the IMS product and service (re-

member, they had been using my money for eight months) that I packaged up the puzzle and sent it back to them, with the proper apology for not filling out the warrantee card because it was one of the things missing from the paperwork. In this letter I told them that I wanted, by return mail, either a full refund, the 3-Meg MM/1 kit that I ordered with full documentation, directions, and pin-outs required to put it together, or a fully assembled 3-meg MM/1 with documentation, pin-outs, etc.

In December I finally got an assembled, 1-Meg MM/1 with exactly the same paperwork package, down to and including NO WARRANTEE CARD! Since I had sent a list of missing items with the returned computer it again shows worse than sloppy workmanship to fail to include even the warrantee card from the list.

During this time there were several other things going on... Paul Ward had called my home twice while I was out on a ship (I am in the Merchant Marine) and my wife asked him why pin-outs were not included in the directions. Paul told her that if I wanted pin-outs I would have to pay extra for them! To which my wife replied with the question, "Do you mean that you have something for sale that is necessary for the operation of the computer that you haven't told Bernie about?" Paul then changed the subject and promised that he would call me back when I returned from the sea. You guessed it -- NO CALL. Current advertising by IMS would lead one to believe that all kinds of software is being developed just for the MM/1. Buy an MM/1 and you can get all these wondrous things. Well, here's a fact. VED/68K was written by Bob van der Poel on a PROTOTYPE 68K board that bears little or no resemblance to the present MM/1 and was put to use by end-users on DELMAR System IV's before it was ever used on a 3-Meg MM/1. And again, the MM/1 was offered for sale in August, 1990. Since I've already waited 10

months and not yet received the 3-Meg MM/1 computer I ordered, I wonder how long the OS-9 community will have to wait for all the wondrous software that IMS is promising us. I am not willing to wait another 10 months for a parallel port and hard drive OR for software, having only the word of IMS that it will be here!!!

Using the MM/1: The configuration they are shipping to customers is this, - 1 Meg RAM, 1 - 1.44 Meg floppy, 1 - serial port, NO SCSI port, NO parallel port, NO expansion slots for modems or RAM expansion. I have tried to backup my distribution disks, - NO GO! The Backup command has no option for single drive backup. I tried to copy them, again, no single drive copy option. O.K. then, how about DSAVE? Again, no single drive option. All my experience with computers tells me not to work with original disks. Since I can't make backups, the MM/1, as it is being delivered, is UNUSABLE!

I can't list a BASIC program to a printer! I can't even write a letter! My son already has TETRIS on his NINTENDO so what good is a 1 Meg MM/1?

From the above limitations I have concluded that using the MM/1 is less satisfying per dollar than a 32K CoCo-2 with a single-sided, 35 track drive.

I have also concluded that someone from IMS assembled the "3-Meg MM/1 with hard drive" on which RAINBOW put it's certification seal. I can't believe an author with any degree of integrity could assemble an MM/1 with an eye toward writing and publishing an evaluation without making a note that the header pins lack proper identification. Come on RAINBOW, "fess up"! Who assembled your machine?

Some specific complaints:

1. I don't like false advertising. The MM/1 was advertised to be able to run MS-DOS and UNIX applications. A letter from Paul K. Ward tells me that it won't be able to do this.
2. I like on time delivery. Can there be any doubt about why I'm ticked

off about this one?

3. The MM/1 was advertised as being able to expand up to 9 Megs of RAM. There are no boards or any way I can see from the literature that this can be done. I will be needing a computer very soon that can use a large amount of RAM or I wouldn't have been looking for anything bigger than my CoCo. Where is the RAM expansion promised?

4. The MM/1, as delivered, is unusable. I say this because the only MM/1 that I've seen is the 1 Meg system with one floppy and no printer port or pin-outs for the 9-pin serial port. It won't even back up a disk. If there is a 3 Megger anywhere in the world that isn't a prototype, would someone please tell me where it is? I have seen no evidence that the MM/1 reviewed/inspected by RAINBOW wasn't a prototype.

5. There are computers on the market (being delivered and used) that do all the things an MM/1 won't do and most of what they claim it will do. Frank Hogg has been selling 68K systems for years. Most people don't buy them because Frank likes them so much he puts a price tag on them that keeps them on his desk instead of someone else's. DELMAR System IV is a "new kid in school" based on an older, well tested 68K industrial computer. Also, the DELMAR sells for the same price as an MM/1 puzzle and has boards available right now for memory expansion, modems, MFM hard drives, and MS-DOS and CoCo compatibility. My complaint is this, - Why should I put up with rotten service, year late delivery, and unusable equipment when there are people willing to deliver a usable product as agreed?

6. I wanted expandability. The MM/1 was and still is advertised as being an expandable machine. IF (capitalized) the 2-Meg I/O board is ever shipped there will be a few more 9-pin serial ports, maybe a centronics parallel port, possibly a game port, and a SCSI port. There are NO OTHER PROVISIONS FOR EXPANTION for ports OR memory! A System IV or Frank Hogg machine has slots which take

any kind of board that will fit them, including more memory, MODEMS, MIDI adapters, SCSI, serial, and parallel ports. THAT is expandability, and the MM/1 just doesn't measure up.

What do I do now?: I told IMS when I sent their puzzle back to them in October that I wanted the 3 Meg kit I ordered or the 3 meg computer assembled or a full refund. They sent what they sent, which is not what I ordered. Since they are not in compliance with the contract I am planning to take copies of our correspondence, their ads, etc. to the U. S. Post Office and turn them over to the Postal Inspectors for investigation and possible prosecution. There are certain laws that must be obeyed when selling by mail order and when advertising thru the mail. I believe IMS to be in clear violation of these laws and it does not seem to bother them in the slightest that they don't deliver goods as promised or refund money paid to them. Maybe a visit from the Postal Inspectors will improve their corporate conscience.

In conclusion: If one's heart is full of Christian charity, each item above that I referred to as being worse than sloppy workmanship might, taken alone, pass itself off as authentic sloppy work. Taken as a group, however, they can not all fit into that category. The business appears to be run by a small group of tinkerers and hackers who have no understanding of, or regard for, good business practice. Since they are continuing the same procedures after being warned by myself and other customers they are displaying an open disregard for anyone's needs but their own.

To those potential OS-9/68K customers out there in the world I recommend this: Buy a System IV if you want a versitile, expandable machine. If you really NEED full 16 bit I/O (System IV has 8 bit I/O) then get one of Frank Hogg's machines. There are also several others on the market so keep your eyes open. Of those MM/1 purchasers who have

had the same experiences I've had I'll ask this: Send me your name, address, phone number, and a letter detailing your complaints about IMS and together we can figure out what we can do about it! We can call ourselves the MM/1 user support group.

This article was written on a CoCo-III with: OS-9 Level II, Window Writer, Multi-View, Burke & Burke RLL hard drive system w/clock, 512K RAM and uploaded with an InfoTel 2400bd modem & OSTerm

Bernie E. Besherse
P. O. Box 9381
Ketchikan, Alaska 99901

Screen Savers Using Shell+/OS-9 Level II

Shutting down and powering up a computer is very hard on it. That's one of the main reasons screensavers were invented, and this shell-script will help you run them. If you have a hard drive, it even parks the drive heads, protecting the drive in case of power loss or surge.

This is a simple Shell+ shell-script that allows you to run various OS-9 level 2 screen savers by choosing the proper number from the menu. It then parks your hard drive. If you don't have a hard drive, or have more than one, you will want to edit the shell-script appropriately.

Simply place 'ss' in your CMDS directory, and you can type ss at any OS-9 prompt. This shell-script REQUIRES Shell+, and, of course, any of the screensavers you want to use.

The screen savers supported are Spiro, Strings, Maze, Busy, and my screensaver version of Landscape. (Landscape2) All are available on Delphi except Busy, which is the Basic09 screensaver in The Complete Rainbow Guide to OS-9 Level II, Volume I: A beginners Guide To Windows. It should be fairly simple to edit the shell-script to add or delete screensavers.

If you are going to be away from

Contd. from Pg. 7

your computer for several hours, shutting it down is probably better for it than leaving it on. However, if you won't be away long, or if background tasks need to run, System Saver may be the best answer.

SYSTEM SAVER SHELL+ SCRIPT:

```

cls
echo System Saver
echo
echo Choose Screen Saver:
echo
echo 1) Spiro
echo 2) Strings
echo 3) Maze
echo 4) Busy
echo 5) Landscape
echo
prompt Choice:
var.1
if %1=1
    (display 1b20 02 0 0 50 18 0 1 1 1b21 </2; spiro ~100
    <>>/1&) >/w

```

```

else
if %1=2
    (display 1b20 08 0 0 28 18 0 0 0 1b21 </2; strings -
    b0 -k -c -t65 ~100 <>>/1&) >/w
else
if %1=3
    (display 1b20 02 0 0 50 18 0 1 1 1b21 </2; maze ~100
    <>>/1&) >/w
else
if %1=4
    (display 1b20 07 0 0 50 18 0 1 1 1b21 </2; busy <>>/
    1&) >/w
else
if %1=5
    (display 1b20 08 0 0 28 18 5 0 0 1b21 </2;
    landscape2 <>>/1&) >/w
endif
clrif
park /dd

```

Questions or comments can be addressed to:
KMTOMPSON on Delphi.

<<< NEW PRODUCT ANNOUNCEMENT >>>

Frank Hogg laboratory and Delmar Company are pleased to announce the addition of G-WINDOWS to their product lines.

"G-WINDOWS is the most powerful windowing package available for OS9/68000." said Frank Hogg, President of Frank Hogg Laboratory, Inc. "With G-WINDOWS you can put OS9/68000 up against any Mac or PC running Windows 3.0 and beat the pants off them. G-WINDOWS puts OS9/68000 and the Tomcat TC70 in the forefront of user friendly computing." he continued. "When G-WINDOWS was introduced at Atlanta this past October, comments like 'better than any competing product', 'fit for a King', 'the best GUI/Windowing package I've seen', etc. were commonly heard. This was because it is exceptionally user-friendly keeping the OS9 operating system hidden from the user while still reaping its many benefits. The more sophisticated user still retains direct access to, and control of, the operating system. G-WINDOWS is a stable product with a proven record of performance in the industrial market and is in use internationally." said Ed Gresick, President of Delmar Company.

G-WINDOWS is now available for the following computers:

- FHL Tomcat TC70
- Delmar System IV
- Gespak computers
- Hazelwood computers

Peripheral Technology computers

G-WINDOWS will be available soon for the following computers:

- FHL QT PC/30 with VGA or SVGA
- OS9000 computers (386 and 486 based)
- MM1

To get more information or to order G-WINDOWS contact:

Frank Hogg
FRANK HOGG LABORATORY, INC.
204 WINDEMERE ROAD
SYRACUSE NY, 13205
VOICE 315/469-7364
FAX 315/469-8537
DELPHI: 'FHOGG'
CIS:70310,317

EDITOR'S NOTE: G-Windows is not a new product. I have been reading reviews of G-Windows in industrial publications supporting OS-9 System Computers ever since 1989. In short, it is very similar to Microsoft's Windows 3.0 with the added features provided by the OS-9 environment.

more on

G-WINDOWS

G-Windows consists of two main parts, the Desktop and Windows. G-Windows requires a 2 button(or more) mouse.

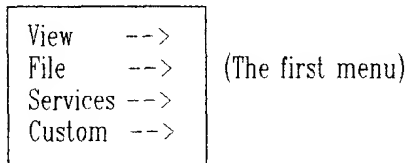
The Desktop handles disks, bit bucket, other things related to manipulating files. A background image can also be displayed. (A girls face is used in the demos.) The following things can be done from the Desktop:

Double click on the left mouse button to do the following:

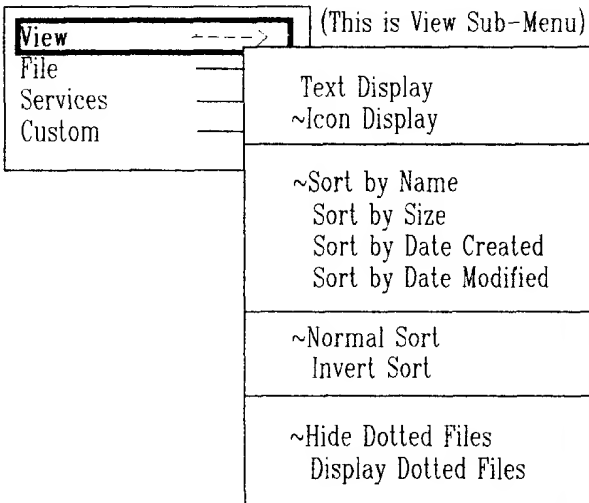
- disk ICON will open the disk. (see window information)
- filename will execute the file.
- directory will open the directory.
- program will execute the program.

Single clicking with the left button will select the item.

Single clicking on the RIGHT button brings up a small menu window with other options right where the mouse is on the screen. These options all have a arrow ('-->') that will when selected bring up a sub-menu window. The following options can be done from this sub-menu. (file options like rename can only be done if a file icon is first selected.) This window looks something like this: (Remember that in G-Windows this would be done with graphics.)



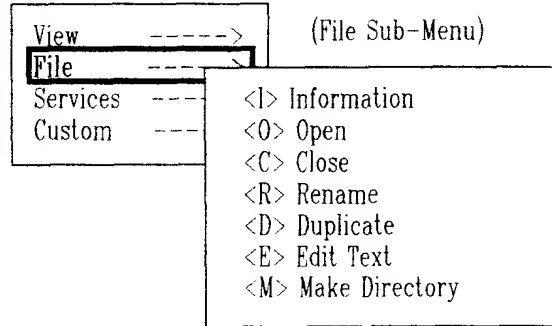
(Move the mouse into the 'View' arrow and...)



The '~' indicates the current mode. (On 'real' G-Windows '~' is a check mark)

In the following menus the letter in the <> such as <I> means that typing 'CTRL I' will be the same as selecting that menu option.

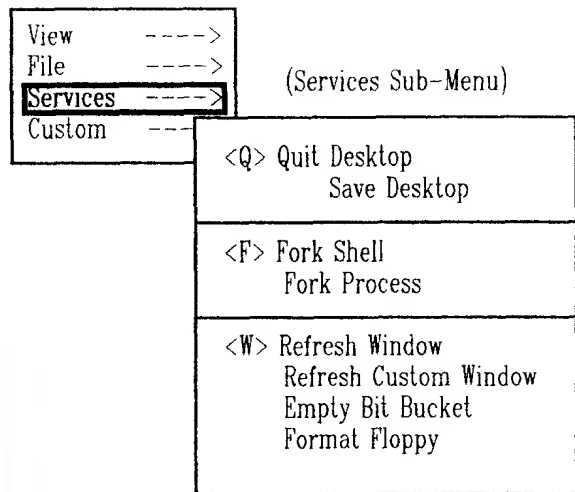
(Move the mouse into the 'File' arrow and...)



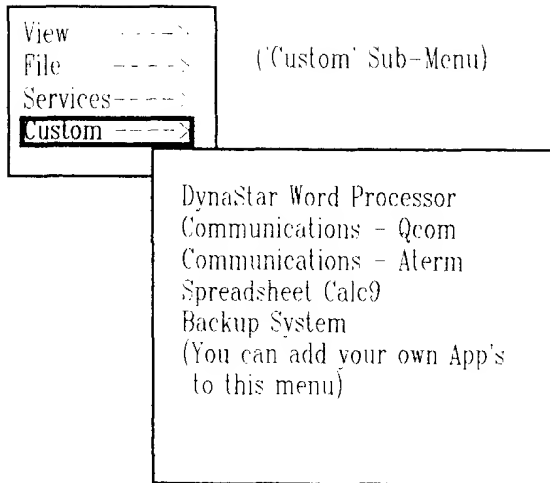
- Information is neat, similar to the info you get from a 'dir -e' but presented in a clearer way.
- Open is the same as double clicking.
- Close is the same as hitting the close button.
- Rename is obvious except slicker, less typing. Typing a 'space' will be replaced by a "... slick!
- Duplicate makes a copy and you get a chance to modify the name it copies it to.
- Edit text calls your favorite editor to edit a text file.
- Make Directory does that.

Many of these options call up what Steve calls an 'alert box'. In Mac parlance this is called a 'dialog box'. This is a box with a area for you to type in plus an OK and CANCEL button. Neat!

(Move the mouse into the 'Services' arrow and...)



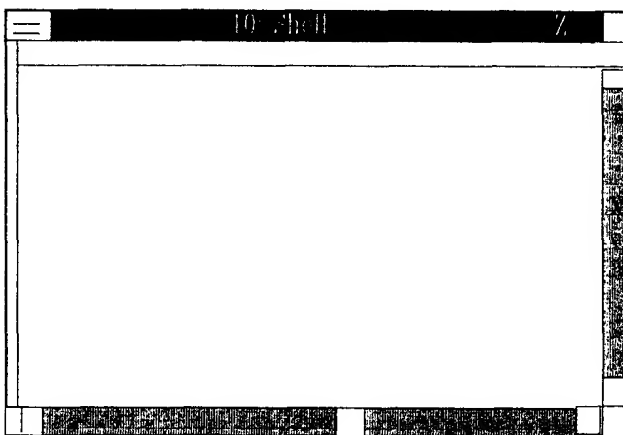
(Move the mouse into the 'Custom' arrow and...)



Windows:

This part of G-Windows is very powerful and complete. It is much easier to use than to describe. A full operational demo disk is available from FHL that will show the complete G-Window package. It has one limitation, it will stop after about 40 minutes. However you can run it as many times as you want by rebooting.

The window is graphical and difficult to describe with text, however the window looks something like this:



All of the window parameters, height, width, displayable height and width, colors, and many other attributes are changeable. Also any number of windows can be started until memory runs out. On a 3.75 meg system this works out to 120+ windows! Some of the more important features are:

- copy and paste between windows. This is part of G-Windows and does not require any modifications to existing applications. We have copied and pasted between editors, word processors, spreadsheets, the shell, all without any difficulty.
- emulates a VT100 terminal.
- drag and resize windows.
- drag and move windows on the desktop.
- reduce windows to a small icon with the 'Z' hibernate button. (reduces screen clutter.)
- shuffle windows on the desktop with the 'move to back' button.

Just some of the things I like to do with windows:

For most work at the shell level I like to create a very long window with 50 lines. I only display 15 or so but I can scroll back to see the previous lines. Combined with copy and paste this is useful.

I use a 'named' window for message sending. For example I put a line in the startup file that sleeps for 35 minutes then puts up a window with the message "G-Windows going down in 5 minutes!" With the demo version this reminds me that it is time to save what I'm doing and reboot.

Named windows are interesting, they are somewhat like named pipes in OSK. Another use is to have a standard place to send error messages too. Different applications can send output to the same named window. One example: Start a long process in a window with this as the last statements in the line: ".echo Done >/win/Process" Then you can hibernate the window and when it gets finished a window will pop up telling you the process is finished! Food for thought huh.

I use the 'Z' (hibernate) button quite a lot. Doing disk formatting for example. After starting the format process I click the 'Z' button. That causes the window to turn into a small icon on the lower right of the screen with a balloon saying 'working'. All I have to do is keep an eye on that 'working' balloon and when it changes to 'waiting' I know the format is done. I then click on the icon and the window reappears.

The move to back button is handy when you have a bunch of windows on the screen. Especially if you have the lower windows covered up and can't find a piece to click on.

Sometimes I start a background process with the output going to a named window. I then shrink that window and put it in the corner. I can keep an eye on it to monitor progress.

I use the copy and paste when I want to do massive editing of text. I bring up the text in one window and copy the pieces to another. This is easier than doing block moves with an editor.

I've been using a Mac since 1984, their window system is the grandfather of them all. Although it and G-Windows are based on the work done by Xerox at Palo Alto the Mac brought the concept to the general public.



Club Activities

Bellingham OS9 Users Forum - Longview/Kelso CoCo Club
Mt. Rainier CoCo Club - Port O'CoCo Club - Seattle 68xxx Mug

Port O'CoCo Club

The Port O'CoCo Club December meeting was extremely well attended with over 20 members present. The A & M Swap meet in Kent was just three days prior to our meeting and the Computer Charity organization handed over all of their CoCo equipment and software to the club in order that two complete systems could be put together and tested and then returned for the Computer Charity for distribution to those in need of a computer but can't afford wone. The remaining hardware and software was then made available to the club to sell at auction during the meeting with the proceeds going to the Computer Charity.

The meeting auction was a great success taking in over \$170 for the Computer Charity, and the club purchased all of the remaining software for \$30. The software will make up a club software library which will be available to club members in January.

The CoCo/OS9 Users Group Banner is finished and was displayed at the meeting. It is approximately 4 x 5 feet, made of vinyl and has a picture of a sailboat with rainbow colored sails with "COCO" and "OS9" printed on the sail. Printed above the sail boat is "SUPPORT YOUR LOCAL USERS GROUP" and beneath the sailboat is the name of all of the current CoCo/OS9 Clubs in Washington. The banner has already been displayed at the December A&T Computer Swap meet in Kent, and is available to any CoCo/OS9 club to display at public functions.

- Donald Zimmerman -



Seattle 68xxxMUG

December 3rd meeting was held at Gugenheim Hall on the University of Washington Campus at 7:30 p.m. The tutorial on writing a Basic09 database was con-

Seattle 68xxxMUG Cont'd

tinued featuring the procedures written by club members. Several Procedures were displayed to show how they functioned and how they were written. Modifications were made to the DELETE and ADD Procedures.

Jeff Brittan's EDIT procedure was a work of art. The source code was approximately 2K and featured a full screen editor.

- Rodger Alexander -



Mt. Rainier CoCo Club

At the December meeting Randy Kirschenmann was elected by a unanimous vote as the new President for 1992. Following the election, new members were introduced: Michael Stokes and John Krisher.

Alan Johnson presented an OS9 catalog program that he wrote in BASIC09. Menus are presented in overlay windows making it easy to use. A printer routine is included to allow any type of printer to use it's capabilities to make an easy-to-read printout. As soon as the documentation is finished, the program will be uploaded to local bulletin boards and will available to members at the next meeting.

Randy Kirschenmann removed the cover from his CoCo tower to give everyone a look at his handiwork. He has created a nice looking package with a 5 1/4in. drive, 3 1/2in. drive, and a hard disk. He explained some of the things that went into putting it together.

The rest of the meeting was spent in discussions and answering some of the many questions of the new members.

At the January 14th meeting, Gary Bondahl will give a presentation of Window Writer, an OS9 word processor. Meetings are held at the Parkland/Spanaway Library, 13718 Pacific Ave. Saurth.

- AlanJohnson -

Cont'd from Pg. 10

I've always considered the Macs windows to be the best because they were designed in. However I've always wished that I had a more OS9 like OS under the Macs windows. Sometimes the Mac can be a pain to work with. G-Windows has all the good features that the Mac has and has OS9 under it for a superb combination. I can do more with G-Windows than the Mac because of OS9. Steve Adams, the creator of G-Windows did an excellent job integrating a better than Mac window interface to OS9. I showed this to a friend who only uses a Mac. He wanted to know why he couldn't do the same things with his Mac!

I vote for G-Windows on the TC70 as 'OSK Product of the Year!' This one thing will do more to bring OSK into the mainstream than any other single software package!

Washington State BBS List

FAR POINT BBS - Seattle

RiBBS (Fido NET)
(206) 285-8335

COLUMBIA HTS. BBS - Longview/Kelso

RiBBS (Fido NET)
(206) 425-5804

DATA WAREHOUSE BBS - Spokane

RiBBS (Fido NET)
(509) 325-6787

BARBEQUED RIBBS - Bellingham

PCBOARD (PC-NET)
(206) 676-5787 Conference #5

OS-9 TACOMA BBS - Tacoma

RiBBS (Fido NET)
(206) 566-8857

COCO EXPRESS BBS - Anacortes

RiBBS (Fido NET)
(206) 293-1057

Bellingham OS-9 Users Group

presents . . .



Written by

Scott Honaker & Rodger Alexander

Mail \$2 + \$1 shipping fee to:
3404 Illinois Lane, Bellingham, WA 98226

The OS-9 NEWSLETTER is published by the Bellingham OS-9 Users Group; Rodger Alexander, Editor. Back Issues are available for 75 cents (please specify month & year).

OS-9 Newsletter

3404 Illinois Lane

Bellingham, WA 98226